

## Lista zadań nr 12

### Zadanie 1. (2 pkt)

Zmodyfikuj maszynę wirtualną z wykładu tak, by nie używała mutowalnego stanu. W tym celu przekazuj stos handlerów wyjątków jako dodatkowy parametr funkcji `exec`.

### Zadanie 2. (1 pkt)

Zaimplementuj w języku MATCH standardowe funkcje `map` oraz `append` operujące na listach.

### Zadanie 3. (1 pkt)

Zaimplementuj w języku MATCH wstawianie liczby do drzewa BST oraz spłaszczanie drzewa do listy. Używając obu funkcji, zaimplementuj sortowanie list.

### Zadanie 4. (2 pkt)

Zauważ, że w OCamlu wzorce mogą pojawić się po lewej stronie równości w konstrukcji `let` oraz jako argument formalny w definicji funkcji. Jest to szczególnie użyteczne, gdy używa się krotek jak w poniższym kodzie.

```
let (min_val, max_val) =  
  List.fold_left  
    (fun (mn, mx) x -> (min mn x, max mx x))  
    (Int.max_int, Int.min_int)  
  xs
```

Zmodyfikuj język MATCH z wykładu tak, by konstrukcje `let` oraz `fun` używały wzorców do wiązania zmiennych.

**Zadanie 5. (2 pkt)**

Zmodyfikuj interpreter języka MATCH tak, by wielokrotne wiązanie tej samej zmiennej w obrębie jednego wzorca powodowało błąd, ale by przesłanianie zmiennych dalej było możliwe. Na przykład próba policzenia poniższego wyrażenia powinna zakończyć się błędem.

```
match (42, 13) with
| (x, x) -> x
end
```

Natomiast poniższe wyrażenie jest poprawne.

```
let fst = fun x ->
  match x with
  | (x, _) -> x
end
in fst (42, 13)
```

**Zadanie 6. (2 pkt)**

W OCamlu gramatyka wzorców dopuszcza konstrukcję `as`. Dopasowanie wartości  $v$  do wzorca  $p$  `as x` skutkuje dopasowaniem wartości  $v$  do wzorca  $p$  oraz związaniem zmiennej  $x$  z wartością  $v$ . Na przykład poniższe wyrażenie policzy się do pary (42, 13).

```
match (13, 42) with
| (x, _) as p -> (fst p, x)
```

Rozszerz język MATCH o podobną konstrukcję.

**Zadanie 7. (2 pkt)**

Popraw implementację funkcji `Eval.string_of_value` w interpreterze języka MATCH tak, by wyświetlana drukowane konstruktory i pary były poprawne składniowo, ale zawierały jak najmniej nawiasów. Na przykład chcielibyśmy drukować

```
C(1, D(), E(F 42))
```

zamiast

```
C(((1, D(())), E(F(42)))).
```